AFOSR-TR-97-0658

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	31 July 1997	1994–1997

4. TITLE AND SUBTITLE

Parallel Program Archetypes

5. FUNDING NUMBERS

Grant FF49620-94-1-0244

6. AUTHOR(S)

Mani Chandy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

California Institute of Technology

Pasadena, California 91125

8. PERFORMING ORGANIZATION REPORT NUMBER

CALTECH-CS-CHANDY-AFOSR-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research

Bolling Air Force Base

nm

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Publically available

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The research supported by this grant falls into three categories: distributed systems, parallel programming, and theory of concurrent compositions. We developed a distributed systems framework, called Infospheres, that allows any Java programmer to create a distributed application in a simple reliable way. The system can be downloaded from the internet at http://www.infospheres.caltech.edu. We built a parallel program framework that helps scientists build mesh on spectral applications for parallel architectures. We proved the correctness of the framework using formal methods.

14. SUBJECT

Software components, mobile agents, program composition parallel composition, formal methods, patterns

15. NUMBER OF PAGES 6

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

FINAL REPORT
AFOSR GRANT FF49620-94-1-0244

Parallel Program Archetypes

Mani Chandy
California Institute of Technology 256-80
Pasadena
California 91125
(626)395-6559
mani@cs.caltech.edu

Summary

The research supported by this grant falls into three categories:

1. Distributed System Archetypes
2. Parallel Program Archetypes
3. Theory of Concurrent Composition

Next, we discuss each of these three categories briefly, and later we describe the research done, papers published and students graduating, in each of these categories in detail.

Distributed Systems

The most successful of the three categories in terms of visibility and papers was distributed systems. Our proposal was to do research on developing archetypal infrastructures that would help others build concurrent systems. The essential idea is that a designer can start with an archetypal program and use it as a framework for developing applications. My group started building a distributed framework called *Infospheres*, and we have made release 1.0 Beta 3, available for downloading from the internet. Several papers on the infrastructure itself, and on distributed applications using the infrastructure have been written.

Parallel Programs

Our research in this area deals with building program archetypes for message passing and shared-memory systems that can be used as a starting point for application development. A specific concern is robustness: we planned to research formal methods of verifying the correctness of program archetypes so that users of these archetypes would have a reliable framework to work with. We have developed and verified a collection of parallel archetypes that have been used to develop parallel

applications by scientists outside the CS department. A PhD dissertation by Ms. Berna Massingill, and one by John Thornley, deal with this subject.

Theory

In this area we are concerned with methodologies for proving the correctness of concurrent composition of objects. As the concept of "software parts" becomes practical and increasingly relevant to the Air Force, the question of systematic ways of plugging parts together becomes important.

A common approach to specifying parts is to use so-called rely-guarantee techniques. We rely on the environment into which a part is placed to satisfy certain properties, and provided the environment has these properties then the part guarantees to have certain properties. The rely-guarantee approach is unsatisfactory for dealing with certain properties because it gives rise to a circular dependency. The environment of a part p is in turn composed of other parts, say q and r; and, the environment for q consists of p and r. So, the specification of a part p depends on p itself.

Prof. Beverly Sanders at the University of Florida at Gainesville and I worked on novel ways of dealing with concurrent composition of parts to avoid this circularity. We have written three papers on this problem for archival journals. (One of the papers is still under review, and the others have been accepted or have appeared.)

Next, we discuss each of these categories in greater detail.

Distributed Systems

Keywords: *software components, software parts, mobile agents, program composition, plug-and-play*

Within a decade we expect that many companies, including Air Force vendors, will build software components by plugging together software parts that are available from the same vendor or other vendors, on the internet. You will be able to put software parts together in two ways: (i) download the code for a part and create an instance of the part yourself, or (ii) plug existing parts together, for instance creating a travel manager part by composing an existing airline ticket reservation part from (say) United Airlines with hotel reservation parts from (say) Holiday Inn and Ramada, and your own calendar appointments scheduler part.

These software parts will be persistent. Your calendar appointments part will persist as long as you do. Software parts may also be mobile. If you get posted to Europe then the parts that help you manage your work will move to Europe with you. Our research dealt with sytematic archetypal ways of composing persistent mobile software parts. Our concept of parts includes many of the models of mobile software agents.

We developed a prototypical infrastructure for creating and supporting software parts. Our AFOSR-supported project is called Infospheres:

http://www.infospheres.caltech.edu

and Release 1.0 Beta 3 has been downloaded by several hundred sites. We are now working on an infrastructure specifically focused on supporting plug and play of software parts. This infrastructure is called the *parts* infrastructure, and we expect it to be released in 1998. The parts infrastructure draws upon many of the ideas in the infospheres infrastructure.

Papers in distributed systems supported by the grant

The ideas underlying Infospheres have been presented in several papers, two of which have received awards. The papers includes the following.

1. K.M.Chandy, A. Rifkin, P. Sivilotti, and others
   "A World-Wide Distributed System Using Java and the Internet"
   Best paper award
   Proceedings IEEE High Performance Distributed Computing 5
   Syracuse, New York, August 1996.
2. K. M. Chandy, and A. Rifkin,
   "Systematic Composition of Objects in Distributed Internet Applications:
   Processes and Sessions,"
   Best paper award Proceedings Hawaii International Conference on Systems Science 30,
   Maui, Hawaii, January 1997.
   Also to appear in Computer Journal October 1997.
3. K.M.Chandy, A. Rifkin, J. Kiniry, D. Zimmerman
   "A Framework for Structured Distributed Object Computing"
   To appear in Journal of Parallel Computing 1998
4. K.M.Chandy, A. Rifkin, J. Kiniry, D. Zimmerman
   "Webs of Archived Distributed Computations for Asynchronous Collaboration"
   To appear in the Supercomputing Journal, October 1997
5. K.M.Chandy, B. Dimitrov, R.Ramamoorthy, and A. Rifkin
   "A General Resource Reservation Framework for Scientific Computing"
   To appear in Proceedings ISCOPE (International Scientific Computing in
   Object-oriented Programming Environments),
   Marina Del Ray, December, 1997.
6. K. M. Chandy and E. Schooler,
   "Designing Directories in Distributed Systems: A Systematic Framework"
   Proceedings IEEE High Performance Distributed Computing 5
   Syracuse, New York, August 1996.
7. L. Thomas, S. Suchter, A. Rifkin,
   "Developing Peer-to-Peer Applications on the Internet:
   the Distributed Editor, SimulEdit
   To appear, Dr. Dobbs Journal, 1997
8. K.M. Chandy,
   "The Scientist's Infosphere"
   IEEE Computational Science and Engineering Journal,
   July 1996.
9. K. M. Chandy, J. Kiniry, and A. Rifkin,
   "The Caltech Infospheres Project"
   OMG/W3C Conference on Distributed Objects and Mobile Code
   Boston, Mass June 96
10. J. Kiniry, and K. M. Chandy,
    "Leveraging the World Wide Web for the World Wide Component Network",
    OOPSLA 1996 Workshop:
    Toward the Integration of WWW and Distributed Object Technology

October 1996, OOPSLA, San Francisco.
11. P. Sivilotti and K. M. Chandy,
"Toward High-Confidence Distributed Programming with Java:
Reliable Thread Libraries"
Proceedings 11th International Conference on Systems Engineering
ICSE 96, Las Vegas, 9-11 July 1996.
12. E.M. Schooler,
"Conferencing and Collaborative Computing",
Proc. Dagstuhl International Workshop on Fundamentals and Perspectives
on Multimedia Systems, pp 175-208
To appear ACM Multimedia Systems Journal
13. (E.M. Schooler) S.Shenker, A. Weinrib, E. M. Schooler,
"Managing Shared Ephemeral State: Policy and Mechanisms",
International Workshop on Multimedia Transport and Teleservices, COST 237
Internet Draft, IETF Multiparty Multimedia Session Control

Students in Distributed Computing

1. Paul Sivilotti will get his PhD in the next term. His thesis deals with using formal methods to help develop practical methods and tools for certifying (i) basic software parts and (ii) composed software parts obtained by plugging simpler parts together. Paul's bachelors degree is from Queen's University, Kingston, Ontario, Canada.
2. Adam Rifkin has completed his Masters thesis. He has worked at the Air Force Laboratory in Rome, New York. He has coauthored many of the papers listed above. He should complete his PhD in early 1999. Adam has a Bachelors and Masters from the College of William and Mary in Virginia.
3. Eve Schooler completed her Masters at Caltech in 1996. She is working on a class of distributed algorithms based on the announce-listen paradigm implemented on top of multicast. Eve now has a Microsoft Fellowship. Eve Schooler has a BS from Yale, a Masters from UCLA and a Masters from Caltech.
4. Dan Zimmerman will be completing his Masters thesis this year. He has now secured an NSF Fellowship. Dan's PhD thesis deals with systematic methods for composing units of workflow. Dan has a Bachelors from Caltech.
5. Joe Kiniry will be completing his Masters this year. Joe's PhD thesis deals with methods of negotiation between software parts. When software parts are produced they may not all be designed with standard pluggable interfaces. What happens if you want to compose two software parts and their interfaces don't quite match? Joe's research deals with automatic negotation between software parts so that "couplers" can be developed between parts with similar --- but not identical --- interfaces. Joe has a B.S. in Mathematics and B.S. in Computer Science from Florida State University at Tallahassee and an M.S. in Computer Science from the University of Massachusetts at Amherst.

Software Produced

Several releases of the Infospheres Infrastructure have been available on the web. The software has a good users guide and we respond quickly to questions and problems.

Outreach

Novell Corporation is interested in exploring collaboration on research in distributed systems, and they generously gave us a gift of $100,000 to further research.

Parasoft Corporation is collaborating with us in two areas: (i) specifying software parts so that programmers can identify parts that they need, and (ii) identifying requirements for Java integrated development environments. They have given us a gift of $25,000 to enable joint research.

Parallel Program Archetypes

Keywords: *Parallel composition, formal methods, message passing, shared memory.*

We have developed a parallel archetype library in C and Fortran to help programmers develop parallel applications. Associated with the library are methods that help programmers develop *sequential* programs that can be converted to parallel programs by systematic transformations that we define. Our approach is different from automatic compiler parallelization because it requires the explicit interaction of the programmer. The programmer manipulates a sequential program through a sequence of steps, planning from the outset to create a parallel program. The parallel library is available from the web.

We have developed a detailed example of a parallel archetype called the mesh-spectral archetype. The mesh-spectral archetype is adequate for developing a variety of parallel applications because it allows both communication between neighbors of cells in a grid, and also communication between rows and columns of cells that are required for spectral computations.

Papers in parallel programming supported by the grant

1. K. M. Chandy, Rajit Manohar, Berna L. Massingill, and Daniel I. Meiron, "Integrating Task and Data Parallelism with the Group Communication Archetype"; Proceedings International Parallel Processing Symposium (IPPS) , 1995.
2. Berna L. Massingill and K. M. Chandy, "Parallel Program Archetypes" Submitted Journal of Parallel Programming

Students in Parallel Programming

Ms. Berna Massingill will complete her thesis by October of this year. She has developed a formal theory of parallel archetypes and program transformations. She has also implemented a parallel archetype library available from the web. Ms. Massingill has worked at the Air Force Laboratories, and has accepted a postdoctoral position at the University of Florida, working with Prof. Beverly Sanders.

Software Produced

The parallel archetype library is available on the web at http://www.etext.caltech.edu/Implementations/ It has been used by Prof. Dan Meiron's group in Applied Mathematics, and for parallelizing an Air Force electromagnetics code.

Theory of Concurrent Composition

This work was done with Prof. Beverly Sanders while she was at Caltech, and later when she went to the University of Florida. The idea of having billions of software parts on the web, that you can mix and match to create new parts, is powerful and compelling. This vision can become a reality only if there are (i) systematic ways of composing parts to get other parts, and (ii) ways of reasoning about a composed part given the specifications, but not the implementations, of its components.

One approach to specifying software parts is by using rely-guarantee specfications that rely on the environment of a part satisfying certain properties and in turn guarantee certain properties of the part itself. As we discussed earlier, this approach can lead to circular dependencies. Prof. Sanders and I have taken a novel approach to rely-guarantee specifications to develop a theory that allows you to reason about the composition of parts in a straightforward way. We have written several papers for archival journals. This work was done largely without students.

Papers in theory of Concurrent Composition supported by the grant

1. K.M.Chandy and B. Sanders,
   "Reasoning about Program Composition"
   Submitted to ACM Transactions on Programming Languages and Systems, 1997
2. K.M.Chandy and B. Sanders,
   "Predicate Transformers for Reasoning about Concurrent Composition",
   Science of Computer Programming, Vol. 24, pp 129-148, 1995.
3. K.M.Chandy and B. Sanders,
   "Predicate Transformers for Reasoning about Concurrent Composition",
   Science of Computer Programming, Vol. 24, pp 129-148, 1995.
4. K.M.Chandy,
   "Properties of Concurrent Programs", *Formal Aspects of Computing Science*, Vol. 6, 1994
5. S. Kryukova, B. Massingill, B. Sanders,
   An Algorithm for Distributed Location Management in Networks of Mobile Computers
   Submitted, Journal of Distributed Computing

This grant has been very successful, particularly in the area of distributed computing. The number of downloads of our software and support from commercial companies are indications of interest in the area.